**用户功能详细设计文档**

**功能一：获取天气信息及锻炼建议**

类：WeatherInformation //根据申请的api获得实时天气信息(实时天气，温度)

具体实现：

|  |  |  |  |
| --- | --- | --- | --- |
| **函数** | **逻辑实现** | 输入数据 | 输出数据 |
| getTodayWeather() | 根据中央气象台的api,建立Connection,获得返回的json格式数据，调用相应的jar包，对该数据进行处理，处理后的数据用map封装，然后返回 | 城市id | Map<String,Object>类型的天气数据 |
| getExeAdvice() | 调用getTodayWeather()的返回值，通过判断语句对相应的天气数据进行判断（map.get(key))，给出对应天气状况下的跑操建议（String） | Map<String,Object>类型的天气数据 | String类型的跑操建议 |

**Addition:对于天气信息的呈现（present()），输入数据为（map.get(“weather”),component），第一个参数为具体的天气，第二个参数为要呈现的component，无具体返回值,逻辑实现:**

**通过判断语句为天气和图标一一对应，再加上具体的温度信息，利用component.set()呈现。**

**功能二：对于学生各类跑操信息的获取及给予建议**

**核心操作：通过点击相应的标签向服务器发送请求，接受服务器回传的数据并处理，最后呈现在跳转的界面上**

**各类跑操信息包括：1，过去跑操的详细数据;2,早晚操的完成情况；3，当天的早晚操情况；4，已经获得的分数/未获得的分数，已经经过的日期/剩余日期**

**其中1，3两点建立在已有日历的基础上**

类：1，MainFrame //各类标签的集合体，包括早晚操完成情况查询，分数/日程进度，日历，跑操建议获取。

2，Calendar //MainFrame的主体,提供对过去跑操信息的查询，当天跑操状况的查询

注：各函数中需要调用一个共有函数ConnectToServer(Map<type,data> map):建立与服务器的连接，向服务器发送信息（请求类型和请求数据内容），并接受服务器的返回数据

|  |  |  |  |
| --- | --- | --- | --- |
| **函数** | **逻辑实现** | 输入数据 | 输出数据 |
| 监听函数1（主要是MainFrame包含的标签，即早晚操的完成情况和跑操建议的获取） | CompletenceOfAP():早晚操的完成情况  调用ConnectToServer(Map<A,time)  (依照约定，服务器应当返回Map<key,data>类型的数据)  对返回的数据调用map.get(key)获取早晚操的完成次数及得分，最后将相应数据通过component.set()进行设置 | Map<A,time> map:  A代表早晚操完成程度的请求类型，time代表希望获得跑操次数的请求  A:id | 早晚操的完成次数及得分 |
| getProAdvice():  获得关于跑操进度的建议，具体实现还没想好 |  |  |
| 监听函数2（主要是在日历上实现的请求，包括历史和当天的跑操情况的请求） | getDailyInfo/getHistoryInfo()  首先获取点击日期date  调用ConnectToServer(Map<B,date>  对返回的数据调用map.get(key)(其中key包括早晚操开始结束时间，跑步路线，平均速度，路程长度，最快速度等）  将解析后的数据在新跳转的界面上呈现:component.set() | Map<B,date> map:  B代表具体日期跑操情况的请求类型，date表示所要请求信息的发生日期  B:id | 早晚操开始结束时间，跑步路线，平均速度，路程长度，最快速度等 |
| PresentProcession():呈现已经获得的分数/未获得的分数，已经经过的日期/剩余日期 | 由于是呈现函数，所以只要将服务器返回的数据填入相应的MainFrame中的component  调用ConnectToServer(Map<C,score>)  建立时间间隔(每隔一天自动去调用这个函数，实现更新效果  将返回的数据呈现在MainFrame的相应component中  component.set()  (需要一点计算) | Map<C,score>):C代表获取已经获得的分数的请求类型  C:id | 返回已经获得的分数/未获得的分数，已经经过的日期/剩余日期（日期可以通过调用系统时间就可以计算） |

**功能三：体育老师发布跑操信息/班主任（辅导员）发送鼓励信息**

**类：Manage（P.Eteacher/monitor):这个功能只是这个类里的一个方法**

|  |  |  |  |
| --- | --- | --- | --- |
| **函数** | **逻辑实现** | 输入数据 | 输出数据 |
| SendMessageAll():向全体成员发布跑操信息  体育老师 | String message;  首先执行I/O操作，输入message;  调用ConnectToserver(Map<D,message>)  (服务器接受请求后就向全体成员发布消息)  ReceiveMessage()&Present():客户端接收信息并呈现(接受的信息类型应该是Map<type(身份），message>  Reflect():服务器返回给体育老师message是否发布成功（boolean)  条件？ | String message  Map<D,message>  D代表发布信息给全体成员的请求类型） | message的呈现  发布信息是否成功（true/false） |
| SendMessageToGroup():向某个班级或某个院系成员发布消息  班主任/辅导员 | String message;  首先执行I/O操作，输入message;  调用ConnectToserver(Map<班级代码/院系代码,message>)  (服务器接受请求后就向班级或院系内的全体成员发布消息)  ReceiveMessage()&Present():客户端接收信息并呈现(接受的信息类型应该是Map<type(身份），message>  Reflect():服务器返回给班主任或辅导员message是否发布成功（boolean)  条件？ | String message  Map<,message>  D代表班级代码/院系代码） | message的呈现  发布信息是否成功（true/false） |

**注：逻辑实现内的某些没有实现的函数其实只是一些操作，可以不写成一个函数**

**注：有些类型虽然有数据库的支持，但是我还是不好确定**